

Proyecto #2 - TableTech

Tecnológico de Costa Rica
Área Académica Ingeniería en Computadores
CE-1103 Algoritmos y Estructuras de Datos I
I Semestre 2023
Valor 20%



Objetivo general

- Implementar un sistema que permita la aplicación del Paradigma Orientado a Objetos mediante la utilización de estructuras de datos jerárquicas.

Objetivos específicos

- Diseñar una solución que permita resolver el problema descrito en esta especificación aplicando patrones de diseño y utilizando el estándar UML.
- Implementar una solución que permita resolver el problema descrito en esta especificación utilizando Programación Orientada a Objetos en Java y haciendo uso de estructuras de datos lineales y jerárquicas.
- Elaborar la documentación correspondiente a la solución implementada para la evidencia del trabajo desarrollado utilizando estándares de documentación técnica y herramientas de gestión de proyectos.

Descripción del problema

La universidad X-TEC ha designado a su equipo la tarea de desarrollar una solución computacional que permita resolver el problema de las largas filas en el comedor institucional. Lo que se requiere es un sistema que permita gestionar los pedidos de los clientes, de manera que se disminuyan las filas en el comedor.

Para esto se requiere que el sistema sea implementado con una arquitectura cliente-servidor, utilizando sockets para la comunicación. A continuación, se describen los elementos que forman parte de la arquitectura de este sistema:

MasterApp

Esta es la aplicación desktop que utilizarán las personas administradoras del comedor institucional. Debe satisfacer los siguientes requerimientos:

- Al iniciar sesión, la persona debe ingresar el username y contraseña.
- Las personas administradoras pueden agregar otras personas administradoras.
- Las personas administradoras también pueden editar o eliminar otros usuarios administradores.
- Agregar platillos al menú, para lo cual debe indicar el nombre, cantidad de calorías, tiempo de preparación y precio.
- Eliminar y editar los platillos del menú.
- Visualizar la cola de pedidos, para lo cual se debe mostrar todos los platillos de cada pedido y el tiempo restante, calculado a partir de la suma del tiempo que preparación de cada platillo. Para esto debe asumirse que la preparación es secuencial, es decir, no puedo iniciar la preparación de un platillo, hasta que el otro se haya terminado.

ClientApp

Esta aplicación de escritorio será utilizada por las personas clientes del comedor institucional. Debe satisfacer los siguientes requerimientos:

- Al iniciar sesión, la persona debe ingresar el username y contraseña.
- Cuando ingresa el usuario, se debe habilitar la opción de visualizar el histórico de los pedidos que ha realizado.
- Permitir la opción de que solo se presenten los pedidos activos, es decir, los que estén en preparación, pendiente de entrega.
- Permitir realizar un pedido, para lo cual debe permitir buscar opciones del menú por el nombre del platillo. Si encuentra el platillo debe mostrar toda la información del mismo y permitir la opción de agregarlo al pedido.
- Una vez que se hayan agregado todos los platillos, se da por finalizado el pedido, y el mismo debe ser incluido en la cola de pedidos del sistema.

NOTA: MasterApp y ClientApp pueden ser implementadas en una sola aplicación, solamente debe asegurarse que, según el tipo de usuario que ingrese, se muestren las opciones de ese tipo de usuario.

ServerApp

Esta aplicación será la encargada de gestionar toda la lógica del sistema, dado que *MasterApp* y *ClientApp* son de despliegue, es decir, solamente despliegan información a los usuarios y permiten la interacción. Lo anterior quiere decir que toda la lógica del sistema debe ser implementada en *ServerApp*, lo cual incluye las estructuras de datos y los algoritmos.

Para soportar los funcionalidades que deben ser implementadas en *MasterApp* y *ClientApp*, la aplicación de *MasterApp* debe implementar lo siguiente:

- La información de todas las personas administradoras debe almacenarse en un único archivo XML. Para la revisión se espera que haya al menos un administrador, en el archivo XML, antes de iniciar la primera ejecución.
- La información de todas las personas clientes debe almacenarse en un único archivo XML. Este archivo debe ser diferente al de las personas administradores.
- Al iniciar la aplicación, debe cargarse, del archivo XML, la información de cada persona administradora y almacenarla en un Árbol Binario de Búsqueda. La validación de la información ingresada por la persona, al iniciar la sesión, debe validarse con la información de este árbol.
- Lo anterior también aplica cuando se elimina o actualiza la información de una persona administradora.
- Al iniciar la aplicación, debe cargarse, del archivo XML, la información de cada persona cliente en un Árbol Binario de Búsqueda. La validación de la información ingresada por la persona, al iniciar la sesión, debe validarse con la información de este árbol.
- Lo anterior también aplica cuando se elimina o actualiza la información de una persona cliente.
- Los diferentes platos del menú deben almacenarse en un archivo JSON.
- Al iniciar la aplicación, debe cargarse, del archivo JSON, la información de cada platillo en un Árbol AVL. Todas las operaciones relacionadas con los platillos deben realizarse en este árbol.
- Cualquier modificación en la información de los árboles debe realizarse también en el archivo, para que en caso de que se detenga la aplicación, la información se mantenga al volver a iniciar.

Controller

Implementado en Arduino. Debe permitir:

1. Visualizar la cantidad de pedidos en la cola
2. Emitir un sonido cada vez que esté listo un pedido
3. Emitir un sonido cada vez que se agregue un pedido. El sonido debe ser diferente al de cuando esté listo un pedido.
4. Cuando un administrador selecciona un pedido en específico, a través de una línea de leds debe indicarse el porcentaje de completitud del pedido. Utilizando la siguiente distribución:
 - a. 1 led = 25% del pedido
 - b. 2 leds = 50% del pedido
 - c. 3 leds = 75% del pedido

Requerimientos no funcionales

1. Cabe resaltar que es de manera obligatoria realizar la maqueta del control para interactuar con la aplicación. No se permite presentar el circuito únicamente en la protoboard.
2. Bibliotecas para comunicación serial con Arduino: jSerialComm o RXTX.

Documentación requerida

1. Internamente, el código se debe documentar utilizando Javadoc. Se debe generar el HTML.
2. La documentación externa se hará en un documento que incluya lo siguiente (deberá entregarse un PDF):
 - a. Breve descripción del problema.
 - b. Diagrama de clases.
 - c. Descripción de las estructuras de datos desarrolladas.
 - d. Problemas encontrados en forma de bugs de *github*: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
3. **Planificación y administración del proyecto:** se utilizará Azure DevOps para la administración del proyecto. Debe incluir:
 - a. Lista de features e historias de usuario identificados de la especificación.
 - b. Plan de iteraciones que agrupen cada bloque de historias de usuario de forma que se vea un desarrollo incremental
 - c. Descomposición de cada user story en tareas.
 - d. Asignación de tareas entre las personas integrantes del grupo.

Aspectos operativos y evaluación:

1. **Fecha de entrega: De acuerdo con el cronograma del curso y lo establecido en el TEC Digital**
2. El proyecto tiene un valor de 20% de la nota del curso.
3. El trabajo es **en grupos de tres personas.**
4. Deben entregar en el TEC Digital un documento con el link del repositorio de GitHub, Azure DevOps y el PDF de la documentación. Para ambas herramientas deben dar acceso al correo del profesor.
5. Es obligatorio utilizar un Git y GitHub para el control de versiones del código fuente y evidenciar el uso de Commits frecuentes.
6. Es obligatorio integrar toda la solución.

7. El código tendrá un valor total de 60%, la implementación del arduino un valor de 20%. La documentación externa y el anexo tienen un valor de 20%.
8. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
9. Se evaluará que la documentación sea coherente, acorde a la dificultad/tamaño del proyecto y el trabajo realizado. Se recomienda que realicen la documentación conforme se implementa el código.
10. La nota de la documentación externa es proporcional a la completitud del proyecto.
11. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
12. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la primera cita de revisión oficial.
13. Aún cuando el código y la documentación externa tienen sus notas por separado, se aplican las siguientes restricciones
 - a. Si no se entrega documentación externa, automáticamente se obtiene una nota de cero en la nota final del proyecto.
 - b. Si no se utiliza un manejador de código se obtiene una nota de cero en la nota final del proyecto.
 - c. Si la documentación externa no se entrega en la fecha indicada se obtiene una nota de cero en la nota final del proyecto.
 - d. Si el código no compila se obtendrá una nota de cero en la nota final del proyecto, por lo cual se recomienda realizar la defensa con un código funcional.
 - e. El código debe ser desarrollado en Java (Windows), en caso contrario se obtendrá una nota de cero en la nota final del proyecto.
 - f. Si alguna persona integrante del proyecto no se presenta a la revisión se asignará una nota de cero en la nota final del proyecto.
14. La revisión de la documentación podrá ser realizada antes, durante o después de la revisión del proyecto.
15. Cada estudiante tendrá como máximo 30 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
16. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
17. Cada estudiante es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión al profesor para coordinar el préstamo de estos.
18. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.

ANEXO DEL PROYECTO

Objetivo General

- Diseñar soluciones creativas para problemas de ingeniería complejos, diseñando sistemas, componentes o procesos para satisfacer las necesidades identificadas con la consideración adecuada para la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como las consideraciones de recursos, culturales, sociales y ambientales según sea necesario.

Objetivos Específicos

- Identificar las necesidades y los requerimientos de un problema complejo de ingeniería considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
- Valorar alternativas de solución para un problema complejo de ingeniería que cumpla con necesidades específicas, considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.

Atributos de Acreditación

- Diseño (Inicial).

Descripción del Entregable

Cada grupo debe elaborar un documento que tenga la siguiente estructura:

1. Portada.
2. Tabla de contenidos.
3. Introducción.
4. Diseño
 - a. Listado de requerimientos descritos con el formato de historias de usuario.
 - b. Describir una alternativa de diseño diferente a la solicitada en la especificación del proyecto.
Para esto se recomienda considerar que este sistema fuera desarrollado para resolver un problema de la vida real, dejando los aspectos académicos que restringen los entregables de este proyecto.

Aspectos operativos y evaluación

1. El documento debe ser enviado en formato PDF, junto con los demás entregables del proyecto.